2 巨大データの扱いと解析

小西 史一
東京工業大学

巨大データとは

生命科学の研究分野において，一言に巨大なデータといっても，数メガバイトのサイズのファイルが数千万個に達するようなデータセットが解析に必要になる場合や，1 つのファイルサイズが数テラバイトになるファイルを作成する必要がある場合など，その研究の方法により構成されるデータセットの形態は多種多様です．一方，その後の解析処理方法に関しては，処理を実行する計算機システムのリソースの限界などから，問題を適宜実行可能な単位に分割して処理するという基本的な方針は，生命科学の中の多くの分野で共通した特徴になっています．多くの研究者は，いきなり規模の大きな解析を実行するのではなく，対象を絞り込み，徐々にスケールアップするアプローチを選択する例が一般的です．ただ，近年これほどデータが大量に出てきたのは，生命現象を明らかにするために必要な計測や測定といった実験の精度の向上と規模の拡大などが一因といえます．

そして，巨大なデータセットの解析を通して研究が進行するデータサイエンスがひとたび本格化してくると，これまで Excel などの表計算ソフトを利用して情報を整理してきた研究者の中には，データの格納や解析といった基本的な研究活動において支障をきたしてしまう状況に陥るケースも出てくるでしょう．

しかし残念なことに，多様化する巨大データのすべてを一元的に，そして最適に扱うことができる解析環境を設計することは一般的には難しいと考えており，だからといって，闇雲に拡大する利用ニーズに合わせて，計算資源やストレージ資源への投資を実施しても，導入後に思ったような効果が得られないことも経験がある方も多いと思います．これは，CPU のマルチコア化や計算サーバのブレード化を背景とした数万 CPU コアに及ぶ高密度なサーバ実装が可能となったことで，計算機システムの設計が，より高い計算処理能力を求める FLOPS 指向へ偏重し，本来性能をバランスさせて設計すべきである，CPU 資源とストレージ資源の性能が，実体の解析処理に合わないことが，主な原因と考えています．

たとえば，これまで，デスクトップ計算機を使って，CPU コア数（通常 1 から 8CPU 程度）に対応して，相同性検索プログラムである BLAST[1] や，ゲノムマッピングのツールである BLAT[2] 等の処理を利用してきた研究者には，数万 CPU コアが実装されているスーパーコンピュータが利用できる場合に，その規模に応じた相同性解析処理を実行できるように思えるかもしれません．しかしながら，その計算機が想定している利用方法から外れる場合には，この期待は裏切られることが多いことを知っておく必要があります．特に入出力ファイルへのアクセスが他の分野のアプリケーションと比較して多い生命科学においては，思った性能が得られない場合に計算機ノードで高速に動作しているはずの CPU が活用されていない状況に遭遇します．この状況は I/O からのレスポンス待ちが発生して，その間 CPU が遊んでいるためです．このような I/O に関する問題は，ストレージに関する研究分野では，よく知られている現象なのですが，現在のように極端に CPU の性能とコア数が増えた現在の計算環境では特に顕在化してきています．我々が活動する生命情報処理の分野において大規模な計算機を利用する場合には，既設の解析環境を利用するので，計算機の性能のバランスに関する諸問題に関しては，特に注意が必要となります．しかしながら，その計算機の設計思想や解析処理の特性をある程度理解することで，解析環境の設計方針を見つけることは可能です．

そこでまずは，その巨大データの発生傾向についてここで考えてみたいと思います．特に最近では，デジタルカメラなどで使われている高密度デバイスを基盤技術とした解析手法が発展していることから，一度に多くの観測データを取得することができるようになりました．たとえば次世代シーケンサの技術では，それぞれのシーケンサ間の違いは存在しますが，DNA に対応した断片に蛍光標識を施し，アレイ状に固定化して光学的に読

み解くという方法が主にとられています．X 線回折像を使ったタンパク質の立体構造解析では，同じく高密度で高精度のディテクタを使います．スイスにある Paul Scherrer Institut の Swiss Light Source（SLS）で開発された PILATUS（pixel apparatus for the SLS）ディテクタ[3]は，6 メガ個の素子（1 枚の画像あたり約 6 メガバイト）で構成されていて，毎秒 10 枚の回折像データを測定することができます（**図 -1**）．このときサンプル結晶をきわめて正確に微少回転させて，さらに高焦点で照射する技術と組み合わせることで，1 サンプル結晶あたり数万枚の回折像を短時間に得て，解像度の高い立体構造の解析を実現しています．このように多くの検知器などの測定装置では，半導体技術を応用した高密度化されたセンサによる技術革新がされていることは疑いの余地はありません．よって，今後もこれまで以上に高密度化されたセンサから得られる大量のデータが高速に産出されるはずです．現在計画されている X 線回折の次期システムでは，数年内にディテクタのサイズを数倍にして，現在の 100 倍以上の枚数の画像を毎秒測定することができるようになると言われています．

また大量のデータが発生するケースとしては，高速化され大規模化した計算機を使ったシミュレーション等の実行が挙げられます．このシミュレーションの過程を記録するためには，数テラバイトのファイルを作成することもあるからです．現在，理化学研究所で推進している次世代スーパーコンピュータ[4]で実現される 10 ペタフロップスもの実効性能を実現するシステムでは，その名の通りペタフロップス規模のシミュレーションを実行することが可能となり，その結果は，エキサバイトのサイズとなるはずです．もちろん，この性能は数万個から数十万個の CPU コアによって実現されるものであるため，記録されるデータファイルも数千から数万個にもなると予想できます．いずれにしても，かなりのデータが生成され，その後の解析が必要とされます．よって，今後も数ペタフロップス級の計算システムが開発されることで，産み出されるデータのサイズは，莫大なものになると認識しなければなりません．ここでは，このような大規模なデータハンドリングにおける諸問題と，そのようなサイズのデータの解析について，現在使われているいくつかの技術について紹介いたします．

巨大データとその扱い

今年の BioHackathon 2009[5]において，先に挙げた大規模データの扱いに対する問題意識から，生命科学に関係する専門家が集まり，BigData に関するサテライトミーティングを行いました．議論の比較的初期の段階から，各自の研究におけるデータのサイズに起因する諸問題について意見交換されました（**図 -2**）．そこで，参加者全員で，各自の持つ生命科学における問題を，2 次元の空間にマッピングを試みました（**図 -3**）．この図は横軸に扱うデータファイルの個数をとり，そして縦軸に，そのファイルのサイズを示しています．この 2 次元の空間上に，生命科学の分野における対象となる課題の名前を置くことで，その扱っている問題の特徴をサイズの個数で捉えるものです．この図に対応するように，現在利用できる技術を重ね合わせることにより，BigData に対して基本的な扱い方の指針を得ることができると考えたからです．

そこでまず初めに生命科学で扱う重要な課題として，タンパク質相互作用（PPI）やゲノム解析（Genome Sequence）や，その配列のアセンブリ（Genome Assem-

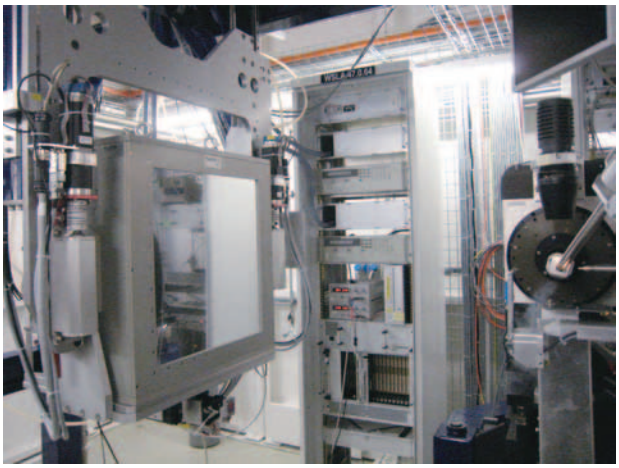

図 -1　Paul Scherrer Institut の Swiss Light Source (SLS) で開発された PILATUS (pixel apparatus for the SLS) ディテクタ．60,000 個の素子（1 枚の画像あたり約 6 メガバイト）で構成されていて，毎秒 10 枚の回折像データを測定することができる．写真の左側の長方形のパネル．

図 -2　BioHackathon 2009 サテライトミーティング BigData における参加者による問題規模のマッピング（中央）．各分野の専門家との共同活動により実現した．規模の大きなデータセットに対するデータアクセスの難しさなどコメントが記述されている．

bly），タンパク質立体構造解析（X-ray diffraction）や，マイクロアレイ（Microarray），そして次世代シーケンサ（NGS）が産出する大量の配列や計算機シミュレーション（MD simulation）といった，大規模なデータを扱うアプリケーションの候補となる分野に関する項目が複数挙げられました．

その選ばれたアプリケーションでは，利用されるデータのサイズや，そのデータセットの規模に応じて図中にマッピングすることができます．たとえばタンパク質相互作用のデータベースのサイズは，数百メガバイトから数ギガ程度であるとのことであり，実験により調べられてきたタンパク質の相互関係のネットワークを表現するため，現状調べられているタンパク質が限られていることなどから，比較的小さいサイズのデータセットとなっています．しかしながら，今後シミュレーションによって推定される相互作用に関する情報なども付加されてくると現状の数百倍の情報量に膨れあがる可能性があります．次にタンパク質立体構造解析の場合には，1 つのタンパク質結晶にごとに 1 回の X 線回折実験で数メガバイトのファイルを数百枚から数万枚の範囲で処理することがあります．また，同時に複数の結晶中のスポットに対して同様の処理を行う場合もあります．したがって，総容量で数ギガから数百ギガバイトのデータセットが 1 つのタンパク質の実験により消費されることになります．そして，マイクロアレイなどに関しても X 線回折実験と同様に CCD 等による画像により，遺伝子の発現の有無を知ることができる原理となっています．よって，スポットの高密度化に伴う，画像解像度の増加傾向があります．また，マイクロアレイでは時系列情報をとることで RNA 発現を捉えるため，時点数や実験の対象数に応じて扱うデータセットの数が増加する傾向もあります．さらに生物の配列情報に関する分野においては，次世代シーケンサとして，新しい原理に基づく配列の読み取りが実現されています．その新しいシステムもこれまで同様に配列に対応した蛍光シグナルを読み取ることで実現されています．ベースコール後のデータがギガバイト程度ではありますが，基となる画像情報を重視する場合には，非常に大きなデータセットを扱うことになります．また，現在多くのゲノムが読み解かれ，さらに個人の情報として蓄積されることも考慮すると，1 億個のデータファイルが作成されることも十分あり得る数字となります．

これまでの実験装置から得られる情報をベースとするデータとは異なり，近年ではシミュレーションにより得られた結果を基に解析を行う例も見られるようになりました．分子動力学などのシミュレーション実験などから得られるトラジェクトリーデータなどは，計算機の性能向上に比例して急速に蓄積されるため，ペタバイトに近いデータサイズが予想されます．

ここで扱った生命科学の課題におけるマッピング結果は，BioHackathon 2009 における議論中に出てきた課題を中心に挙げてあり，厳密な規模の調査により決定したものではありませんが，おおむね現状を表しているものとしています．また，それぞれの研究の方向性や解析処理の進め方によっては，1 つのファイルのサイズが巨大化する場合や，データセットの数が増加する解析などが可能です．したがって，今後は研究の進展によっては，現在では空いているスペースに拡大することが予想されます．ここでより重要なのは，このような空間に与えられた課題の特性をマッピングすることで，現在，もしくは将来使えるデータの格納や解析技術にマッピングすることにあります．

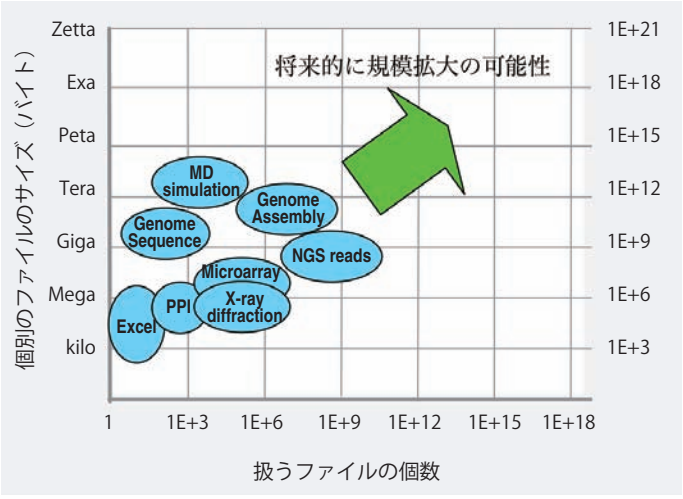

図-3　生命情報処理の課題におけるファイルサイズと数の関係

データ格納について

巨大な情報に対する解析の実行には，そのデータの永続的な形での保存が不可欠となります．特に，生命科学においては，データを保存し第三者が検証可能な形で公開することでサイエンスとして発展してきている経緯があるからです．また DNA のような生命の記憶媒体と対照されることからも，データの格納が生命科学にとって重要であることが分かります．つまり，あらゆる生命体が持つゲノムを格納する技術を，現在においても我々は持ち合わせてはいませんし，すべての個体差を記述するためのスペースも持っていません．その情報のほんの一部でも理解できる形で記録するためには，現在利用できる格納技術について知っておく必要があります．

■ ハードディスクドライブ

そこで，次に先ほど作成した図 -3 に対応する形で，現在利用することができるデータ格納に関係する技術を，ファイルの数とサイズという 2 次元の特性から，図 -4 のようなグラフを作成することができます．

このグラフで最も，内側にあるのは，ハードディスクドライブ（HDD）と呼ばれる磁気ディスクの単体での構成を示しています．その容量は，現在のところ 3.5 インチのサイズで，約 1 台あたり数百ギガバイトから 2.0 テラバイトに達しています．容量に関していえば，CPU と同様に非常に早いペースで増加していますが，そのアクセス速度に関していえば，現在流通が確認されているものでは，4,200・5,400・7,200・10,000・15,000rpm のように，数倍の回転数に差がついている程度となっていることが分かります．HDD は内部に磁性体を塗られたプラットと呼ばれる回転体を持ち，ヘッドと回転の精密な制御により実現されている磁気記憶装置です．容量の大きなディスクは比較的アクセス速度が遅い傾向があり，2 テラバイトの SATA ディスクでは，約 5800rpm の回転数となっています．これは，容量の大きなディスクでは，より多くの枚数のプラッタを重ねて記憶容量を確保していることとプラッタ径を大きくとるために，回転数が低く設定されているからです．したがって，逆に SCSI 用のディスクなどでは，高性能な 10000rpm 以上のものが採用されていますが，プラッタの枚数を少なくすることで容量は比較的少なく設定されています．いずれにしても，ランダムなアクセス速度は，このディスクの回転速度に大きく影響を受けることになります．生命科学分野における処理では，複数の CPU コアからの独立したプロセスからの同時ファイルアクセスなど，そのディスクに対するアクセスパターンは通常ランダムになります．したがって，ランダムアクセスの性能の向上は，生命科学分野における貢献は高いと見ることができます．よって，より高速なランダムアクセス性能と，大容量の HDD を生命科学の分野では求めています．

■ RAID ディスク構成による HDD 集約

先に紹介した HDD の速度と容量を同時に増加させる方法としてディスクを複数接続してディスクアレイを実現する方法の 1 つに RAID という複数のディスクを仮想的に 1 つのディスクに見せる技術があります（図 -5）．RAID0 や RAID5 などは一般的によく使われる設定で，ストライピングと呼ばれる複数台のハードディスクに対して，ファイルを分散して読み書きする方法があります．この技術によりファイルの断片を並列にアクセスすることができるために高速な I/O 性能を実現することができます．解析データの格納場所として利用する場合に，ファイルのサイズが 1 つの HDD のサイズを超える場合や，規模の大きなデータセットを扱う場合には，RAID でアレイ構成されたボリュームを利用するのが一般的になっています．したがって，図 -5 のように 1 台の計算機内で閉じた形で，数台の HDD で構成された RAID ディスクであれば，高速な I/O 特性を実現することが比較的容易で，マルチコア環境下であっても優れたパフォーマンスを期待できます．さらに，RAID1 のミラーリングや，RAID5 などの冗長化構成が存在しており，ディスクの破損によるデータの消失を防ぐ工夫がされていることも重要です．

■ NFS によるデータ共有

大規模なデータを複数台の計算機によって共有して，そのデータ解析処理を並行して実行することで，処理能

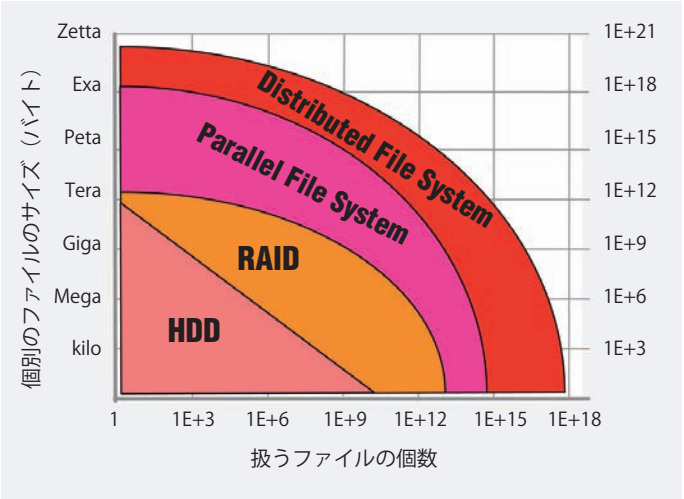

図 -4　データ格納技術におけるファイルサイズと数の関係

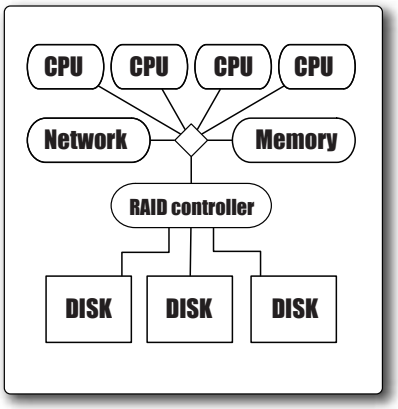

図 -5　RAID コントローラを備えたデスクトップ

力を向上させる技術として Beowulf 型のクラスタ計算機[6]が知られています（図-6）．その中核を成す技術として NFS（Network File System）と呼ばれるファイル共有サービスがあります．また，この NFS の機能を専用化して，ネットワークから利用できるようにしたシステムは Network Attached Storage（NAS）と呼ばれています．特にファイルシステムと計算機ノード間を専用のネットワークとして構成されたものを Storage Area Network（SAN）とも呼んでいます．この場合，ストレージ関連の通信が，他のノード間のコミュニケーションによる通信と分けることができるために，安定した高性能な運用が可能となります．いずれにしても，このような NFS での構成における問題点は，計算ノード数が多い場合には NFS 間のネットワークのボトルネックが発生しやすいことが問題です．通常の回避策としては，ボリュームを提供する NFS サーバを複数台用意して，負荷分散を図る方法や，一斉にアクセスしないなどの運用上の対策がとられています．

■ 並列ファイルシステムによる共有ファイルシステムの実現

多数の計算機ノードからの読み書きの要求に応えるために，ファイルサーバ側もクラスタ構成することで，高い I/O 性能を実現することができます．

NFS などのサービスを提供してきたストレージシステムを，I/O サーバとして，複数台でクラスタ構成させることで，より大規模で高い性能のストレージサービスを提供する仕組みとして，並列ファイルシステムが存在します（図-7）．現在主に使われている並列ファイルシステムには，Sun Microsystems の LusterFS[7] や IBM の GPFS, そしてオープンソースプロジェクトで開発されている PVFS2[8] 等のほか数種類が知られており，その構成はおおむね，I/O サーバとなるストレージを持つノードを複数ネットワークで接続し，それらの I/O サーバの情報を管理するメタサーバが介在し，ファイルシステムには専用のクライアントソフトウェアによりアクセスすることで実現されています．この I/O サーバ間ではおおむね RAID と同様のストライピング機構がノード間で実装されていることで高速化されます．このような並列ファイルシステムを利用することで，I/O サーバの実装されているディスク総容量よりも，さらに大きな 1 つのファイルを作成することも可能となります．この大規模な並列ファイルシステムが，稼働している例としては，東京工業大学の TSUBAME スーパーコンピュータ[9]をはじめ，国内でも筑波大，東大，京大の 3 大学に設置された T2K オープンスパコン[10]などがあります．TSUBAME の場合では，1 ノードあたり 48 個の 500 ギガバイト容量の HDD を搭載し RAID 構成した I/O ノードを 60 台で

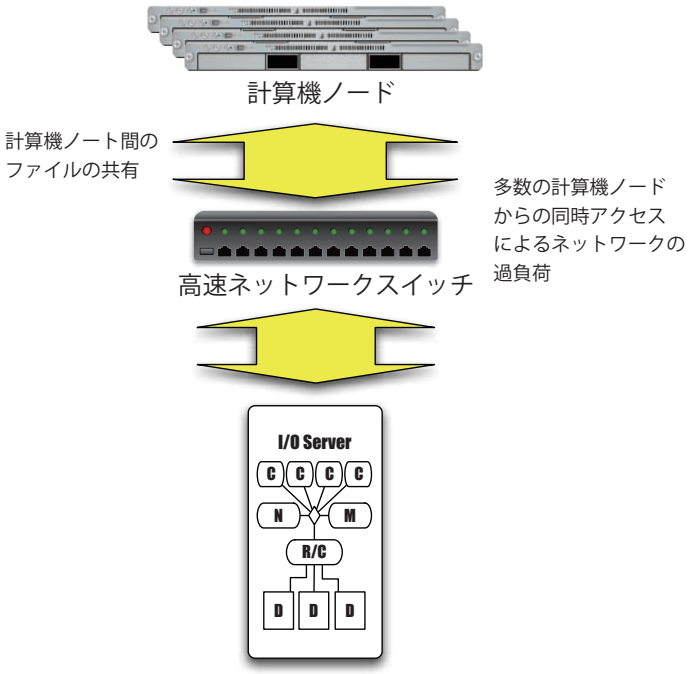

図-6　NFS によるデータ共有モデル．複数の計算ノード間の情報共有化による並列情報処理の実現．多数のノードからのファイルサーバへの同時アクセスによるネットワークおよび，ディスクアクセスへの過負荷による性能の低下．

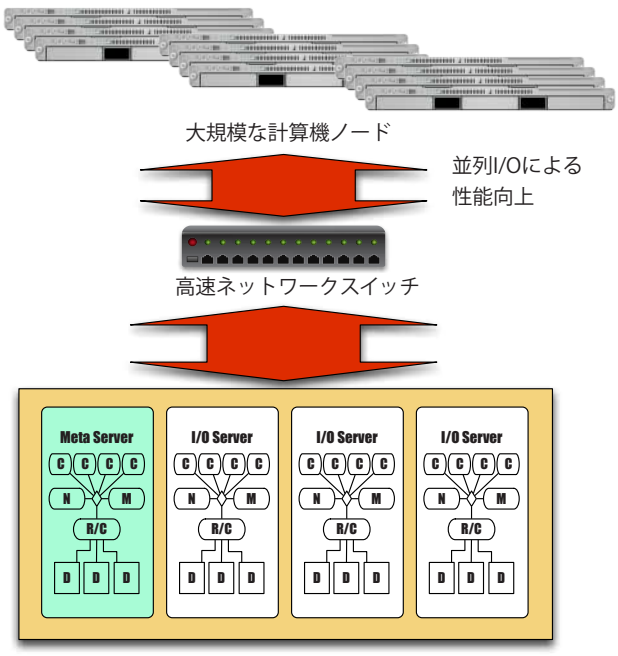

図-7　並列ファイルシステムによるストレージクラスタ．複数の I/O サーバへの同時アクセスによるストライピングアクセスの実現による I/O 性能向上の実現．ファイルの更新や，ディレクトリ作成などのメタデータ更新に関する操作に対する過負荷には注意が必要．

構成して，Luster FS のボリューム総量 1.5 ペタバイトを実装しています．そして，その I/O 総合帯域幅の性能は 60GB/s を達成しています．しかしながら，このシステムをもってしても 10480 CPUcores/655nodes からの同時アクセスを想定しているわけではありません．並列ファイルシステムでは，比較的サイズの大きなファイルのハンドリングにシステムが調整されていることが多く，サイズの小さいファイルや，ディレクトリ構造が深く複雑な場合などには，思ったほどの性能を発揮することができません．並列ファイルシステムを利用しないクラスタ計算機での例として，理化学研究所において運用されていた RSCC（Riken Super Combined Cluster）システム[11] では，ジョブの投入時に計算ノードのローカルストレージに指定したファイルをステージングする機構が実装されていました．しかしながら，ステージングされていた各計算ノード上のボリュームは，1 台，もしくは 2 台の HDD で構成されており，RAID による十分な高速化がされてはいませんでした．したがって，各ノード上で実行しているプログラムは，高速な I/O を利用することができず，そのノードでの性能は通常の計算機と同等または，それ以下になってしまうケースがアプリケーションの種類によっては知られていました．そのようなアプリケーションの特性は，BLAST などの生命情報処理で使われているアプリケーションに共通しています．たとえば，分割可能な大量の問合せクエリーに，大きなサイズのファイルを走査して，その結果を出力するタイプのものです．また，このように並行して実行された結果を，回収する処理も無視できません．このような不具合は，計算機システムの設計時にデータインテンシブなジョブの実行に関する意識が，不十分であることを意味しています．

■ 分散ファイルシステム

並列ファイルシステムと分散ファイルシステムの一番の違いは，そのシステムに格納されているファイルの局所性を活かした機能が実装されているかの違いです．特に，国際間でのデータ共有や，ネットワーク遅延が長い環境下においては，並列ファイルシステムのような方法では，常にネットワークに負荷がかかり，実際の運用に耐える速度で利用することができないことが予想されます．もちろん，このような用途に耐え得るようなネットワーク機器を開発するなども考えられますが，通常それは莫大な費用がかかりコストパフォーマンスがよくありません．そこで，分散ファイルシステムは，アクセスされるデータの局所性を活かして，近いクライアントからアクセスさせる発想で設計されています．もう少し踏み込めば，データの存在している局所的な環境で，そのデータを必要とするプログラムを実行できるようにします（**図 -8**）．もちろん，NFS や並列ファイルシステムで実現してきた，ファイルの一元的で透過的なアクセス機能はそのままで，同様に利用できます．また，これまで並列ファイルシステムでは，複製に対する扱いが十分ではありませんでしたが，分散ファイルシステムでは，任意の数の複製を効率的に作成する機能も持っています．計測器からのデータ産出の際にも，利用できるストレージノードにファイル単位で均等に格納できます．その結果，計算機とストレージが一体となり，そのノード数を増やしても，ノードの局所的な I/O 性能を維持でき，結果的にノード数に対してスケールアウトするストレージ環境を実現できます．この場合，ファイルの実体が収まっているボリュームは，RAID などのディスクアレイ技術により十分に高速化されていることが必要となります．このようなファイルシステムを実現したのが，Google 社のシステムを構成している GoogleFS[12] や，Yahoo! などのシステムを構成している Hadoop File System，そして Gfarm[13] です．さらに，このような環境下において MapReduce のような計算フレームワークを組み合わせることで，大規模なデータ処理環境を実証しています．よって，今後は巨大なデータ処理には不可欠なストレージとして注目されています．

大規模データ解析について

大規模データの解析の例として，次世代シーケンサから得られる核酸配列をアミノ酸配列データに照会する作業を TSUBAME スーパーコンピュータ上で実施する際に実際にとられる方法を紹介いたします．ただし，TSUBAME は非常に多くの利用者がいることと，ストレージ関しては共用されているため性能評価が難しいなどの点から，性能に関する数値情報は提示しないことをあらかじめご理解いただきたいと思います．

たとえば，以下のような次世代シーケンサから得られる規模での一般的な配列の情報処理を実行するというケースがあります．この場合相同性検索とは，DNA を構成する ATGC の組合せから構成される文字列を，紹介先の配列集合に生物的な評価尺度において似ているかを調べる作業となります．

事例：問合せ核酸配列　1,000 万本（75bp）をアミノ酸配列　500 万本（約 3 ギガバイト）に対して，BLAST による相同性検索の照会を行う．

生命科学における大規模な処理のうち，より小さい問題に分割しやすい問題と，分割しにくい問題があります

が，事例としてあげた相同性検索の問題は，問合せ配列の数がきわめて多いにもかかわらず，配列ごとに独立した問合せという点では，分割しやすい課題といえます．極端な例では 1,000 万台の計算機を使えば，1 回の BLAST 検索を実行することで完了することができるからです．しかしながら，実際には 1,000 万台の計算機を利用することはできませんし，結果を集める労力を無視することもできませんので，たかだか数百個の規模の CPU 資源を使って結果を出さなければなりません．

BLAST の場合には，I/O 性能の低下が無視できるのであれば，多くの CPU 資源にジョブを割り振る方が効果的です．したがって，経済的に確保することができる最大の CPU 数を使うのが最初のストラテジーとなります．

さて，最大利用可能な CPU 数を使う際に気をつけなければならないのは，その BLAST ジョブの実行性能がノードに比例してスケールアウトするかという点にあります．もし並行して実行している他の BLAST ジョブの影響をストレージのレスポンスを通して受ける場合には，同時に走らせているジョブ数を減らすなどの処置をすることで，影響を少なくする必要があります．また，他のユーザが実行しているジョブの影響なども考えられますので，その場合には，より影響の少ない別の領域のストレージへの移行が必要となります．

TSUBAME でも通常計算ノードから直接読み書きを推奨しないホーム領域を使用した場合には，他のユーザに対して影響を与えるような現象が容易に起こります．高性能な LusterFS 上であってもアクセスするノードの規模や，アクセスのパターンによってはさらに大きな影響を受けますが，計算ノードが持つ固有のローカルストレージ領域を使うなど，LusterFS を使わない処置を行う必要が出てきます．現在の TSUBAME ではローカルディスク領域は提供されていませんが，適切なマナーのもとで /tmp などの特殊な領域を使用することで回避することは可能です．経験値的に I/O 性能低下を計る指標としては BLAST であればプロセスの CPU 使用率が 20%を切るようであれば，I/O の影響を疑う必要があります．また，LusterFS では，ファイルの Open と Close を繰り返すような操作や，多数のディレクトリの作成と削除を行う作業を多数実行することは一般的に苦手とされますので，BLAST 検索の後処理として，スクリプト言語などで処理を行う場合にも注意が必要です．

今回の場合には，問合せ先のデータベースのサイズが

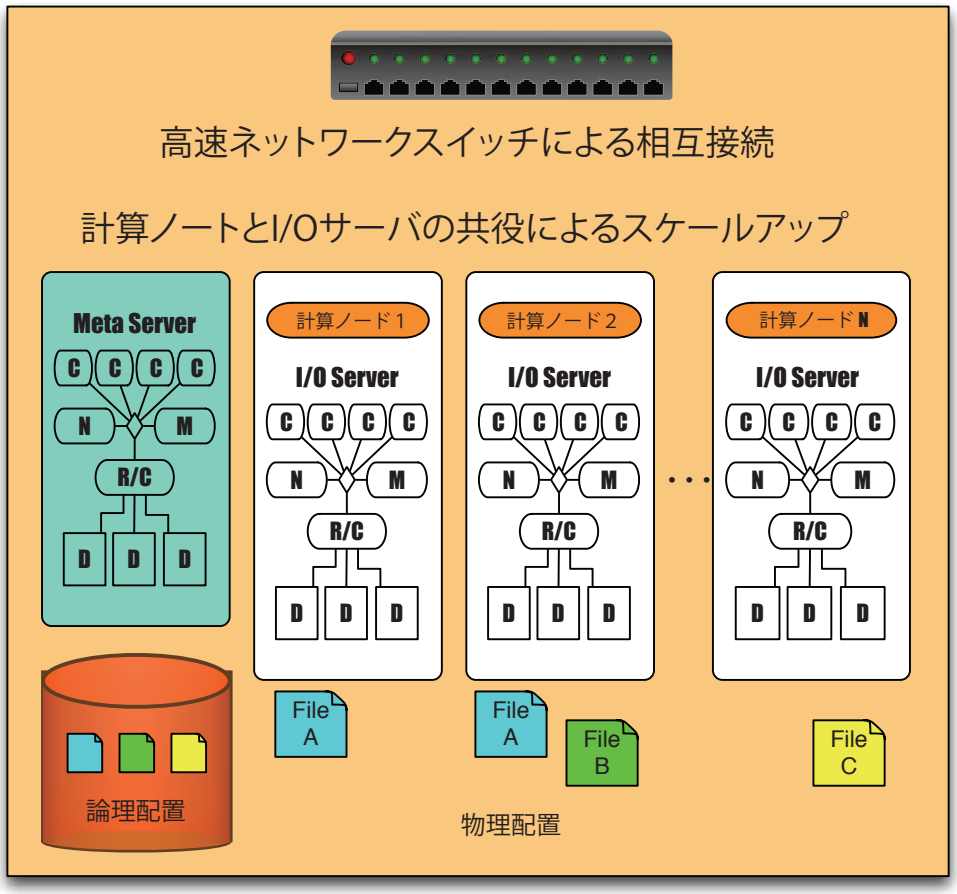

図-8　分散ファイルシステムによる構成．計算機ノードとI/Oサーバを同一ノードで共役させることで，スケールアウトするファイルシステムの構成を実現．各ノードには論理的なファイル配置情報が提供されNFSと同等の利便性が提供される．また物理配置としては，File Aの用に複数のノード上にファイルが分散されることでファイルアクセスを局所に抑えることができる．

3 ギガバイト程度と比較的サイズが大きいこともあり，結果として得られる出力にアライメントを含めないとしたら，ストレージの負荷を抑えることができます．通常の計算ノードでは 32 ギガバイトのメモリを実装しており 3 ギガバイト程度のデータベースであれば，キャッシュの効果を期待することができます．LusterFS を構成しているシステムは 60 台のサーバで構成されており，各サーバノードに実装されている CPU の数からも，その 4 倍の 240 の CPU コアによる実行が望ましいと考えています．実際には，ネットワークインタフェースなどの要素もあり，数倍の余裕があると考えられますが，他のユーザの使用を加味すると 200CPU が安定して利用することができる経験則になっています．よって，5 万本の問合せ配列を，200 個に分散されたジョブにより実行を試すことになります．結局これは，スケールアウトするシステム構成にするためには，計算ノードと同じ数のストレージノードが望ましいということになり，ボトルネックに CPU 資源を合わせているにほかなりません．

データファーム構築の勧め

これまでスーパーコンピュータといえば，演算速度を中心とした設計がされ，Top500 リストの LINKPACK のスコアがどの程度出ているかが，計算機としての評価の軸となっていました．この傾向はしばらく続くものと思っていますが，私は，このような演算速度をもとに設計するのではなく，CPU と I/O のバランスで計算機を設計する時代が来ていると考えています．今後さらに性能向上して高密度化する計測機器を使って，分子からエコシステムまでの非常に幅の広い範囲でのデータを中心とした科学が構築されようとしており，そのような新しい科学を推進するためには，演算能力だけの向上では意味がなく，I/O 性能を含めた上での性能評価がなされなければ，国際的な競争から一歩も二歩も遅れることになります．データファームというシステムの構成が，これまで以上に認知され，データインテンシブなジョブの実行にふさわしい環境の整備が生命科学を支える原動力になるはずです．

参考文献

1）Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. : Basic Local Alignment Search Tool, J. Mol. Biol., 215, 3:403-10 (1990).
2）Kent, W. J. : BLAT — the BLAST-like Alignment Tool, Genome Res., 12, 4:656-64 (2002).
3）PILATUS 6M, http://pilatus.web.psi.ch/pilatus.htm
4）次世代スーパコンピュータ開発実施本部，http://www.nsc.riken.jp
5）BioHackathon 2009, http://hackathon2.dbcls.jp/
6）Becker, D. J., Sterling, T., Savarese, D., Dorband, J. E., Ranawak, U. A. and Packer, C. V. : BEOWULF : A PARALLEL WORKSTATION FOR SCIENTIFIC COMPUTATION, Proceedings, International Conference on Parallel Processing (1995).
7）Luster File System, http://www.sun.com/software/products/lustre/
8）PVFS, http://www.pvfs.org/
9）TSUBAME スーパーコンピュータ，http://www.gsic.titech.ac.jp
10）T2K Open Supercomputer, http://www.open-supercomputer.org/
11）RSCC, http://w3cic.riken.go.jp/rscc/
12）Ghemawat, S., Gobioff, H. and Leung, S.-T. : The Google File System, In Proc. of the 19th ACM Symposium on Operating Systems Principles (SOSP'03), pp.29-43 (Oct. 2003).
13）Gfarm Datafarm - Gfarm file system, http://datafarm.apgrid.org/

（平成 21 年 7 月 14 日受付）

小西 史一（正会員）
konishi@is.titech.ac.jp

2001 年東京都立科学技術大学　博士（工学）．1998 年東京都立科学技術大学助手，2000 年理化学研究所研究員，2008 年東京工業大学グローバル COE「計算世界観の深化と展開」特任准教授．専門は，大規模バイオインフォマティクス．